AMENDMENTS

In the Claims

The following is a marked-up version of the claims with the language that is underlined ("___") being added and the language that contains strikethrough ("—") being deleted:

1. (Currently Amended) A method comprising:

training an email system for determining spam, where training includes at least the following:

retrieving a first email message;

generating a phonetic equivalent of at least one word from a body portion the email message;

tokenizing the phonetic equivalent of the word to generate a token representative of the phonetic equivalent;

tokenizing at least one word in a subject line of the first email message;

tokenizing at least one simple mail transfer protocol (SMTP) email address associated with the first email message;

tokenizing at least one domain name associated with the first email message;

tokenizing at least one attachment of the first email message, wherein tokenizing the at least one attachment includes generating a 128-bit MD5 hash of the attachment, appending a 32-bit length of the attachment to the generated MD5 hash resulting in a 160-bit number, and UUencoding the resulting 160-bit number;

determining a spam probability from the generated tokens;

in response to a determination that the spam probability from the generated tokens indicates that the first email message is likely spam:

determining whether the generated tokens are present in a database of tokens;

in response to a determination that at least one of ~~the,~~ the generated tokens is not present in the database of tokens, assigning a probability value for each token as spam and adding the token and assigned probability value to the database of tokens; and

in response to a determination that the token is present in the database of tokens, updating a probability value of the token;

in response to a determination that the spam probability from the generated ~~tokens,~~ tokens indicates that the first email message is not likely spam:

determining whether the generated tokens are present in a database of tokens;

in response to a determination that at least one of the generated tokens is not present in the database of tokens, assigning a probability value for each token indicative of non-spam and adding the token and assigned probability value to the database of tokens; and

in response to a determination that the token is present in the database of tokens, updating a probability value of the token;

sorting the generated tokens in accordance with the corresponding determined spam probability value; and

filtering a second email message according to the training.

2. (Previously Presented) The method of claim 1, wherein generating the phonetic equivalent of the word comprises:

identifying a string of characters, the string of characters including a non-alphabetic character; and

removing the non-alphabetic character from the string of characters.

3. (Previously Presented) The method of claim 2, wherein removing the non-alphabetic character comprises:

locating a non-alphabetic character within the string of characters, the non-alphabetic character being at least one selected from the group consisting of:

" (quote);

' (single quote);

! (exclamation mark);

@ (at);

(pound);

$ (dollar);

% (percent);

^ (caret);

& (ampersand);

* (asterisk);

((open parenthesis);

) (close parenthesis);

_ (underscore);

- (hyphen);

+ (plus);

= (equal);

\ (backslash);

/ (slash);

? (question mark);

(space);

(tab);

[(open square bracket);

] (close square bracket);

{ (open bracket);

} (close bracket);

< (less than);

> (greater than);

, (comma);

: (colon);

; (semi-colon);

and . (period).

4. (Previously Presented) The method of claim 1, wherein determining the spam probability comprises:

assigning a spam probability value to the token; and

generating a Bayesian probability value using the spam probability value assigned to the token.

5. (Previously Presented) The method of claim 4, wherein determining the spam probability further comprises:

comparing the generated Bayesian probability value with a predefined threshold value.

6. (Previously Presented) The method of claim 5, wherein determining the spam probability further comprises:

categorizing the email message as spam in response to the Bayesian probability value being greater than the predefined threshold.

7. (Previously Presented) The method of claim 5, wherein determining the spam probability further comprises:

categorizing the email message as non-spam in response to the Bayesian probability value being not greater than the predefined threshold.

8. (Currently Amended) A training email system for determining spam on a computer storage medium comprising:

means for receiving an email message having a word and an attachment;

means for generating a phonetic equivalent of at least one word from a body portion of the email message;

means for tokenizing the phonetic equivalent of the word to generate a token representative of the phonetic equivalent;

means for tokenizing at least one word in a subject line of the first email message;

means for tokenizing at least one word in a subject line of the first email message; tokenizing at least one simple mail transfer protocol (SMTP) email address associated with the first email message;

means for tokenizing at least one domain name associated with the first email message;

means for tokenizing at least one attachment of the first email message, wherein tokenizing the at least one attachment includes in generating a 128-bit MD5 hash of the attachment, appending a 32-bit length of the attachment to the, generated MD5 hash resulting in a 160-bit number, and UUencoding the resulting 160-bit number;

means for determining a spam probability from the generated tokens;

in response to a determination that the spam probability from the generated tokens, means for indicating that the first email message is likely spam:

means for determining whether the generated tokens are present in a database of tokens;

means for, in response to a determination that at least one of ~~the,~~ the generated tokens is not present in the database of tokens, ~~means for~~ assigning a probability value for each token as spam and adding the token and assigned probability value to the database of tokens; ~~and~~

means for, in response to a determination that the token is present in the database of tokens, ~~means for~~ updating a probability value of the token; and

means for, in response to a determination that the spam probability from the generated ~~tokens, means for indicating~~ tokens indicates that the first email message is not likely spam:

determining whether the [[,]] generated tokens are present in a database of tokens;

in response to a determination that at least one of ~~the,~~ the generated tokens is not present in the database of tokens, assigning a probability value for each token indicative of non-spam and adding the token and assigned probability value to the database of tokens; and

in response to a determination that the token is present in the database of tokens, updating a probability value of the token;

sorting the generated tokens in accordance with the corresponding determined spam probability value; and

filtering a second email message according to the training.

9. (Currently Amended) A system comprising:

a processor; and

a memory, the memory storing:

receive logic configured to receive an email message having a word and an attachment;

phonetic logic configured to generate a phonetic equivalent of the

word from the email message;

first tokenize logic configured to tokenize the phonetic equivalent of the word to generate a token representative of the phonetic equivalent; and

second tokenize logic configured to tokenize the attachment;

tokenizing at least one word in a subject line of the first email message;

tokenizing at least one simple mail transfer protocol (SMTP) email address associated with the first email message;

tokenizing at least one domain name associated with the first email message;

tokenizing at least one attachment of the first email message, wherein tokenizing the at least one attachment includes in generating a 128-bit MD5 hash of the attachment, appending a 32-bit length of the attachment to the, generated MD5 hash resulting in a 160-bit number, and UUencoding the resulting 160-bit number;

determining a spam probability from the generated tokens;

in response to a determination that the spam probability from the generated tokens indicates that the first email message is likely spam:

determining whether ~~the,~~ the generated tokens are present in a database of tokens;

in response to a determination that at least one of the generated tokens is not present in the database of tokens, assigning a probability value for each token as spam and adding the token and assigned probability value to the database of tokens; ~~and~~

in response to a determination that the token is present in the database of tokens, updating a probability value of the token; and

in response to a determination that the spam probability from

the generated ~~tokens,~~ tokens indicates that the first email message is not likely spam:

determining whether the generated tokens are present in a database of tokens;

in response to a determination that at least one of ~~the,~~ the generated tokens is not present in the database of tokens, assigning a probability value for each

token indicative of non-spam and adding the token and assigned probability value to the database of tokens; and

in response to a determination that the token is present in the database of tokens, updating a probability value of the token;

sorting the, generated tokens in accordance with the corresponding determined spam probability value; and

filtering a second email message according to the training.

10. (Previously Presented) The system of claim 9, the memory further storing:

string-identification logic configured to identify a string of characters, the string of characters including a non-alphabetic character; and

character-removal logic configured to remove the non-alphabetic character from the string of characters.

11. (Previously Presented) The system of claim 10, the memory further storing:

spam-probability logic configured to assign a spam probability value to the token; and

Bayesian logic configured to generate a Bayesian probability value using the spam probability value assigned to the token.

12. (Previously Presented) The system of claim 11, the memory further storing:

compare logic configured to compare the generated Bayesian probability value with a predefined threshold value.

13. (Previously Presented) The system of claim 12, the memory further storing:

spam-categorization logic configured to categorize the email message as spam in response to the Bayesian probability value being greater than the predefined threshold.

14. (Previously Presented) The system of claim 12, the memory further storing:

spam-categorization logic configured to categorize the email message as non-spam in response to the Bayesian probability value being not greater than the predefined threshold.

15. (Previously Presented) A computer-readable medium that includes a program that, when executed by a computer, causes the computer to perform at least the following:

receive an email message having a word and an attachment;

generate a phonetic equivalent of the word from the email message;

tokenize the phonetic equivalent of the word to generate a token representative of the phonetic equivalent;

tokenize the attachment;

generate a phonetic equivalent of at least one word from a body portion of the email message;

tokenize the phonetic equivalent of the word to generate a token representative of the phonetic equivalent;

tokenize at least one word in a subject line of the first email message;

tokenizing at least one simple mail transfer protocol (SMTP) email address associated with the first email message;

tokenize at least one domain name associated with the first email message;

tokenize at least one attachment of the first email message, wherein tokenizing the at least one attachment includes in generating a 128-bit MD5 hash of the attachment, appending a 32-bit length of the attachment to the, generated MD5 hash resulting in a 160-bit number, and UUencoding the resulting 160-bit number;

determine a spam probability from the generated tokens;

in response to a determination that the spam probability from the generated tokens, indicate that the first email message is likely spam:

determine whether the, generated tokens are present in a database of tokens;

in response to a determination that at least one of the, generated tokens is not present in the database of tokens, assigning a probability value for each token as spam and adding the token and assigned probability value to the database of tokens; and

in response to a determination that the token is present in the database of tokens, updating a probability value of the token; and

in response to a determination that the spam probability from the generated tokens, indicates that the first email message is not likely spam:

determining whether the generated tokens are present in a database of tokens;

in response to a determination that at least one of the, generated tokens is not present in the database of tokens, assigning a probability value for each token indicative of non-spam and adding the token and assigned probability value to the database of tokens; and

in response to a determination that the token is present in the database of tokens, update a probability value of the token;

sort the generated tokens in accordance with the corresponding

determined spam probability value; and

filter a second email message according to the training.

16. (Currently Amended) The computer-readable medium of claim 15, the program further causing the computer to perform at least the following:

~~computer-readable code adapted to~~ instruct a programmable device to identify a string of characters, the string of characters including a non-alphabetic character; and

remove the ~~non-alphabetic~~ non-alphabetic character from the string of characters.

17. (Previously Presented) The computer-readable medium of claim 15, the program further causing the computer to perform at least the following:

assign a spam probability value to the token; and

generate a Bayesian probability value using the spam probability value assigned to the token.

18. (Currently Amended) The computer-readable medium of claim 17, the program further causing the computer to perform at least the following:

~~computer-readable code adapted to~~ instruct a programmable device to compare the generated Bayesian probability value with a predefined threshold value.

19. (Previously Presented) The computer-readable medium of claim 18, the program further causing the computer to perform at least the following:

categorize the email message as spam in response to the Bayesian probability value being greater than the predefined threshold.

20. (Previously Presented) The computer-readable medium of claim 18, the program further causing the computer to perform at least the following:

categorize the email message as non-spam in response to the Bayesian probability value being not greater than the predefined threshold.